

### **Amendments to the Claims**

1. (currently amended) A method of translating computer program code from a first language representation into a second language representation, the method comprising:

translating translatable instructions of an input stream in a first language representation into an output stream in a second language representation;

identifying an unresolvable code ~~translation error~~ in the input stream wherein a translator is unable to translate the unresolvable code from a first language representation into a second language representation; and

placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable code ~~translation error~~ in the input stream;

wherein the placed at least one second language representation instruction is at least one of either a handling instruction or an exception throwing instruction.

2. (currently amended) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where the unresolvable code ~~translation error~~ in the input stream would have been placed in the output stream had the unresolvable code ~~translation error~~ been a translatable instruction.

3. (currently amended) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where a method containing the unresolvable code ~~translation error~~ in the input stream would have been placed in the output stream had the entire method been translatable.

4. (currently amended) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where a basic block containing the unresolvable code ~~translation error~~ in the input stream would have been placed in the output stream had the entire basic block been translatable.

5. (original) A method of executing the output stream of claim 1, the method comprising:

executing at least one translated instruction and at least one placed second language representation instruction.

6. (original) The method of claim 1, wherein said placing further comprises:  
directing the placement of the at least one second language representation instruction within the output stream based on a declarative textual indication contained in the input stream.

7. (original) The method of claim 1, wherein said placing comprises placing both:  
at least one exception throwing instruction; and  
at least one handling instruction.

8. (previously presented) A method of executing the output stream of claim 7, the method comprising:  
executing the placed handling instruction subsequent to executing the placed exception throwing instruction.

9. (currently amended) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace an unresolvable code translation error.

10. (currently amended) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace a basic block containing an unresolvable code translation error.

11. (currently amended) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace a method containing an unresolvable code translation error.

12. (original) The method of claim 1, further comprising:  
determining from a declarative textual indication in the input stream which at least one second language representation instruction to place in the output stream.
13. (previously presented) The method of claim 1, further comprising:  
obtaining from a library of available at least one second language representation instruction, the at least one second language representation instruction placed in the output stream.
14. (original) The method of claim 1, further comprising:  
wherein the at least one of either a handling instruction or an exception throwing instruction, is an application programming interface instruction to a dynamically linkable library.
15. (currently amended) A method of translating computer program code from an input stream in a first language representation into an output stream in a second language representation, and the input stream may or may not be from a trusted source, the method comprising:  
translating translatable instructions of the input stream into the output stream;  
identifying suspected code and unresolvable code translation errors in the input stream;  
upon determining by a translator, that the unresolvable code in the input stream cannot be translated from a first language representation into a second language representation, placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable code translation error in the input stream in the first language representation;  
determining that the input stream is from a trusted source; and  
translating the suspected code in the input stream into the output stream.
16. (original) The method of claim 15, wherein a declarative textual indication in the input stream is used to determine whether the input stream is from a trusted source.

17. (previously presented) The method of claim 15, wherein the determining further comprises:

making a request to a server; and

determining from a server response whether the input stream is from a trusted source.

18. (original) The method of claim 5, wherein the presently executing instruction is the at least one placed second language representation instruction, the method further comprising:

requesting by the presently executing at least one placed second language representation instruction, the request being made to a server for instructions external to the output stream.

19. (original) The method of claim 1, further comprising:  
determining a level to place the at least one second language representation instruction;  
wherein the determination of level to place the at least one second language representation instruction is made from among a set of available levels, the set of available levels including at least two distinct levels from a group of potential levels, the group of potential levels comprising a method level, an instruction level, a basic block level, and a program level.

20. (original) The method of claim 19, wherein a declarative textual indication indicates the level to place the at least one second language representation instruction.

21. (currently amended) A method of translating computer program code from an input stream in a first language representation into an output stream in a second language representation, the input stream comprising declarative textual indications, the method comprising:

translating translatable instructions in the input stream into the output stream;

identifying a first language representation of a declarative textual indication in the input stream, the declarative textual indication indicating how to handle an unresolvable instruction ~~translation error~~ encountered in the input stream wherein an unresolvable instruction in the input

stream is an instruction a translator is unable to translate from a first language representation into a second language representation in the output stream; and

translating the first language representation of the declarative textual indication in the input stream into the second language representation of the declarative textual indications in the output stream;

whereby the second language representations of the declarative textual indications are available to a next phase of translation, the next phase of translation able to use the second language representation of the declarative textual indication as a resource for determining how to handle an unresolvable instruction translation error encountered by the next phase of translation as the next phase translates the output stream into a third language representation.

22. (currently amended) A method of translating the output stream of claim 21 into a third language output stream, the method comprising:

translating translatable instructions of the output stream into the third language output stream;

identifying an unresolvable instruction translation error in the output stream and determining that a translator is unable to translate the unresolvable instruction;

determining an indicated third language representation instruction from a declarative textual indication in the output stream; and

placing the indicated third language representation instruction in the third language output stream responsive to identifying the unresolvable instruction translation error in the output stream.

23. (canceled)

24. (currently amended) A method of translating computer program code from an input stream in a first language into an output stream in a second language, the method of translating including multiple sub-units of translation, the method comprising:

translating translatable portions of the input stream into the output stream;

identifying a suspected code in the input stream, determining by a security sub-unit of translation based on a declarative textual indication in the input stream that the input stream is from a trusted source, and translating the suspected code into the output stream;

identifying a first unresolvable instruction ~~translation error~~ in a first method in the input stream of the first language wherein one or more sub-units of translation are unable to translate the first unresolvable instruction into the output stream in the second language, determining by a basic block identification sub-unit of translation, that a second language instruction inserted in the output stream in place of the first unresolvable instruction ~~translation error~~ can not be safely segregated within a determinable basic block within the first method in the output stream, and invoking a method level exception throwing instruction insertion sub-unit of translation to insert an exception throwing instruction in the output stream, and invoking a handler data structure insertion sub-unit of translation to insert a handler data structure in the output stream; and

identifying a second unresolvable instruction ~~translation error~~ in a second method in the input stream of the first language wherein one or more sub-units of translation are unable to translate the second unresolvable instruction into the output stream in the second language, determining by the basic block identification sub-unit of translation, that a second language instruction inserted in the output stream in place of the second unresolvable instruction ~~translation error~~ can be safely segregated within a determinable basic block within the second method in the output stream, and invoking a basic block level exception throwing instruction insertion sub-unit of translation to insert an exception throwing instruction in the output stream, and invoking the handler data structure insertion sub-unit of translation to insert a handler data structure in the output stream.

25. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) for translating translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying an unresolvable instruction translation error in the first language input stream wherein an unresolvable instruction is an instruction a translator is unable to translate from a first language representation into a second language representation; and

instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable instruction translation error in the first language input stream.

26. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) translating translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying suspected code and an unresolvable code translation errors in the first language input stream wherein instructions translating translatable portions of the first language input stream are unable to translate the unresolvable code into the second language output stream;

instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable code translation error in the first language input stream;

instruction(s) determining the source of the first language input stream; and

instructions(s) translating the suspected code into the second language output stream in response to determining that the first language input stream is from a trusted source.

27. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) for translating translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying an untranslatable instruction unresolvable translation error in the first language input stream; and

instruction(s) placing a handling instruction in the second language output stream in response to identifying the untranslatable instruction ~~unresolvable translation error~~ in the first language input stream wherein an untranslatable instruction is an instruction in a first language input stream that a translator is unable to translate into an instruction in a second language output stream.

28. (currently amended) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to perform some portion of the translation process; and

a translation sub-unit operational to ~~translate~~ determine that code ~~an unresolvable translation error~~ identified in the first language input stream can not be translated from a first language into a second language, and responsive to the determination, inserting into an inserted exception throwing instruction in a ~~translated code of~~ the second language output stream.

29. (currently amended) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to ~~translate an unresolvable translation error encountered in the first language input stream into an~~ inserted a handling instruction in a translated code of the second language output stream wherein the handling instruction is inserted in the second language output stream subsequent to a translation sub-unit being unable to



translate unresolvable code encountered in the first language input stream into code in the second language output stream; and

a translation sub-unit operational to verify that a source of the first language input stream is a trusted source, and translating a suspected code encountered in the first language input stream into a translated code in the second language output stream upon verifying that the source of the first language input stream is a trusted source.

30. (currently amended) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to perform some portion of the translation process; and

a translation sub-unit operational to determine that ~~translate~~ an instruction unresolvable translation error encountered in the first language input stream cannot be translated into the second language output stream, and in response to the determination, inserting a into an inserted handling instruction in a translated code of the second language output stream.

31. (previously presented) The computer readable medium of claim 25, wherein the first language input stream further comprises a declarative textual indication indicating where in the output stream to place the exception throwing instruction.

32. (previously presented) The computer readable medium of claim 27, wherein the first language input stream further comprises a declarative textual indication indicating where in the output stream to place the handling instruction.

33. (currently amended) A computer readable medium comprising translated code produced by a computer performing a method comprising claim 1.

34. (currently amended) A method of delaying identification of compile time errors until run time, the method comprising:

compiling a first file containing a first language representation of a computer program into a second file containing a second language representation of the computer program;

identifying an ~~unresolvable~~ instruction in the first file that a compiler is unable to translate into a second language representation;

determining where the identified ~~unresolvable~~ instruction in the first file would be placed in the second file by the compiler if the identified ~~unresolvable~~ instruction were ~~translatable~~resolvable;

~~translating the identified unresolvable instruction into a second language representation of an exception throwing instruction;~~ and

inserting ~~the~~ a second language representation of ~~the~~ an exception throwing instruction into the second file near where the identified ~~unresolvable~~ instruction would have been placed in the second file by the compiler if the ~~unresolvable~~ instruction were ~~translatable~~resolvable.

35. (original) The method of claim 1, further comprising:

placing in the output stream an unaltered copy of the identified ~~unresolvable code~~translation error.

36. (original) A method of executing the output stream of claim 35, the method comprising:

executing the placed at least one second language representation instruction, said executing invoking a translator on the unaltered copy of the identified ~~unresolvable code~~translation error from the output stream, thereby causing the invoked translator to attempt to translate the unaltered copy of the ~~unresolvable code~~translation error into a second language representation instruction, and if the unaltered copy of the identified ~~unresolvable code~~translation error

~~translation error~~ is then translatable into a second language representation instruction, then the second language representation instruction is executed.

37. (original) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a new basic block in the output stream.

38. (original) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction should be inserted in a new basic block.